Journal of Engineering and Development, Vol. 13, No. 4, Des (2009) **ISSN 1813-7822**

Proposed Protocol for State Tampering Mobile Agents Security

Ass. Tech Media Abdul razak ali
Al-mustansiriya University
College of Engineering
Computer & Software Eng.Dept
swenmedia@yahoo.com

ABSTRACT

Several advantages have been identified in using mobile agents in distributed systems .The most frequently advantages include: reduction of network load ,decrease in communication latency, dynamic adoption and better support for mobile devices with intermittent connection .However the benefits offered have not been sufficient to simulate their wide spread deployment, the main reason is their inherent security risks. Security issues consist of protecting the agent platform and protecting the mobile agent. The toughest task is protecting the mobile agent, which is subject to attacks from the platform it is operating on. This paper introduces a protocol based on publicly verified signature chain, digital envelope and centralized Trusted Third Party (TTP) scheme that allows detection of attacks against code, state and execution flow of mobile agents.

Keywords: *mobile agents security, state tampering, execution tracing, publicly verified signature chain, digital envelope.*

الخلاصـــــة

توفر انظمة الوكيل الجوال العديد من الايجابيات في مجال الانظمة الموزعة. اهم هذه المزايا هي تخفيض الحمل على الشبكة،تقليل زمن الاتصالات ،التكيف الفعال والدعم الافضل للاجهزة الجوالة ذات الاتصال المتقطع. بالرغم من تقديمها لهذه المزايا الا انها لم تكن كافية لكي تكون واسعة الانتشاروالسبب الرئيسي في ذلك هي المخاطر الامنية المرافقة و التي تشمل حماية الوكيل الجوال و حماية الوسط العميل. تعتبر حماية الوكيل الجوال من الوسط العميل المتطلب الاساسي في هذا النظام لذا يقدم هذا البحث بروتوكول لحماية الوكيل الجوال باستخدام تقنية سلسلة التوقيع الموثقة عموميا ، المغلف الرقمي و الطرف الثالث الموثوق به مركزيا .الحماية تشمل البرمجة ، البيانات و تسلسل التنفيذ.

1. Introduction

Mobile agents are identified as the basic platform for future frameworks of distributed electronic services. They can be defined as programs that migrate from a host to another to achieve a task specified by them owner. A mobile agent can interact with the components or resources on the platforms if it gets their approvals. It can also communicate with other agents on a platform if the platform's agent environment supports multiple agents. The mobile agent paradigm possesses many advantages; however, security is one of the major obstacles that prevent the large-scale deployment of mobile agent systems. Since the codes of mobile agents are executed on remote platforms rather than only on the home platform, security concerns arise to protect the agents if the remote platforms are malicious. On the other hand, there are threats to security if the mobile agent is malicious. It may attack the hosts (platforms) which enable it to execute [1, 2].

2. Mobile Agents Versus Client/Server Architecture

Initially the mobile agents technique generated as an approach to solve the limitation of the client-server model, in which the client is limited to the operations provided at the server. If the client needs a service that a particular server does not provide, it must find a server that can satisfy the request by sending out more messages to other servers.

This clearly is an inefficient use of network bandwidth. In addition, this kind of communication may increase the networks traffic, waste network bandwidth and causes delay of the reply due to server down time or crashes. Another problem is in the use of the model for disconnected computing. Due to the mobility factor of mobile devices (mobile phones, Personal Digital Assistant, Laptops), sporadic disconnection is frequent in the wireless environment. Mobile agents provide a solution for the dynamic environment of the mobile devices because they do not rely on server operations. The mobile agent appears to tackle significant problems whether in wired or wireless communication such as disconnection operations, increased network traffic and others. Once the mobile agent has migrated, the connection between client and server will be disconnected. This saves network bandwidth, especially in a wireless environment. When a mobile agent finishes its job at the server, it will then be ready to reconnect to its host or to migrate to another node to perform other duties within the network [3].

3. Security

Security is an important issue for mobile agent applications, especially for electronic commerce. The security issues is divided into two sub-issues: host security and agent security. Host security refers to problems about malicious mobile agents. They may migrate to remote hosts and intrude, for example, changing system settings, steal private information,destroy important information of the hosts, etc. Trojan Horses are well-known attack programs . Since the hosts may continuously accept agents and execute them, one may keep sending mobile agents to explore security loopholes of the hosts. The host security problems also exist in other mobile code systems, therefore they have been well-known since the early times.

Popular solution approaches include authentication , authorization and verification of code integrity[4].

Agent security refers to problems about malicious hosts. Since the agent is executed in the environment provided by remote hosts, the hosts can take complete control of the code, states and data of the agent. If the system is closed and all hosts are trusted, the agent should be safe. However, if the system is open and the host is not trusted, the host may be malicious and attack the agent. Major attacks include: unauthorized modification of code, extraction of valuable data, denial of execution, and execution tampering [5]. This paper focuses on solving the problem of execution tampering.

4. General Protection schemes

To solve the execution tampering problem, there are three main broad approaches:

4-1 Trusted Execution Environment:

This approach ensures that the execution environment for the agent to execute is known to be safe. In this way, the agent needs not to worry about any attacks. A closed system is an example. All hosts in the system are trusted. The utilization of a trusted hardware computing base can also provide a trusted execution environment. Unfortunately, many mobile agent systems are not closed system[6],trusted hardware method has weaknesses similar to the closed system approach. It is harsh for all hosts to have to install the trusted hardware in an open system. This limits the number of hosts the agents can travel to even if the hosts are not malicious [7].

4-2 Tamper-Prevention:

The object of this approach is to make tampering of agents difficult. To achieve the goal, many different techniques have been suggested. For example, black box technique that generates an agent abstraction in which the code cannot be read by the host even during execution. An agent is a black box if the code and data of the agent specification cannot be read nor modified at any time, Unfortunately, there is a fatal restriction of this approach: currently only polynomial and rational functions can be used for this approach. This is also the restriction of the encrypted functions, computing protocol programs are needed to be checked before using this approach. However, it is very rare to see programs which consist only of polynomial and rational functions. This restriction makes the approach unrealistic. Another method is the mess-up algorithms, mess-up algorithms can be used to generate a new agent out of an original agent so that they are different in code and data representation but produce the same results. This can increase the attack costs if a malicious host has already known the original agent specification. However, counter

attacks exist against the algorithms. They can work effectively if the choice of mess-up algorithm is known [8].

Another tamper-prevention method is introducing noise code .This approach is to disrupt the malicious host when it tries to locate important code of the agent. There are also some limitations to this approach First, the noisy code (dummy code) would decrease efficiency of the agent by increasing computation time and resources. The penalty increases if stronger protection is to be achieved. Second, a malicious host may simply modify all the codes which are likely to perform important tasks. In this situation, this approach has no protection effect[9].

4-3 Tamper-Detection:

This approach is to detect whether a mobile agent has been attacked along its journey. The most commonly used method is publicly verifiable chained digital signature protocol.

The publicly verifiable chained digital signature protocol [10] uses the data encapsulation technique where an offer is encrypted with some other information to create the encapsulated offer. This protocol relies on a public key infrastructure.

Each platform in this system has a unique identifier and a public signature verification key that is issued by the certification authority. There also exists a directory service from which to retrieve certificates. In this protocol, each platform signs with its secret key for non-repudiability and unforgability. They also encrypt the offer with their public key for data confidentiality. The protocol uses the hash chaining mechanism to link the offer from the current platform with the identity of the next platform. The hash chain prevents a server from modifying its own offer later. In this protocol, anyone is allowed to authenticate the servers involved, thus publicly verifiable.

The protocol begins with the originator P_0 picking a random number r_0. He then hashes r_0 with the identity of the next platform P_1, producing the chaining value H_0. Then he creates the encapsulated offer O_0 by signing the encryption of his dummy offer o_0 and r_0 with the hash value H_0. The encapsulated offer O_0 is sent to platform P_1. P_1 uses the encapsulated offer he received to compute his chaining value, i.e. hashes the encapsulated offer with the identity of the next platform. He then creates his encapsulated offer in the same way as before. The equations are given as follows:

Encapsulated offer: $O_i = SIG_{Pi}(ENC_{P0}(o_i, r_i), H_i)$

Chaining relation: $H_0 = h(r_0, P_1)$

$H_i = h(O_{i-1}, P_{i+1})$

In [11], it is explained that the protocol has a weakness. An adversary can truncate offers by removing them from the end of the set, build its own chain of offers, and submit them to the mobile agent. This weakness would cause the originator into believing that the mobile agent collected these offers while in fact it never even visited the platform.

5. Proposed Security Scheme

The designed protocol is based on three techniques, publicly verifiable chained digital signature protocol , centralized trusted third party and digital envelope scheme. The first one was introduced in the previous section. The centralized TTP is used to authorize and keep track of migration paths.

The third scheme is used to overcome the biggest limitation of public key encryption techniques, is that they are typically much slower to compute than secret key encryption techniques of comparable security. Therefore, a popular scheme is the digital envelope in which the message proper is encoded using a fast secret key technique, and the much shorter key for the encrypted message is then encoded using a public key technique, with the public key of the intended recipient, both being embedded in the single transmission. The recipient will then use his or her private key to decrypt the secret key, and then use the secret key to decrypt the body of the message.

The protocol consists of two parts a registration phase and migration phase. The first one begins by the originator of the mobile agent sending a TTP a registration message. Assuming a safe link with the TTP, the message consists of the signature of originator identity and the hash of agent code and state and list of itinerary (Ih) which contains the list of platforms the mobile agent should migrate to.

$P_0 \rightarrow$TTP: SIGpo(P_0, h(C,S), Ih)

The TTP verifies the signature and returns the originator the agent token which is the signature of agent originator identity , the hash of its code(C) and state(S) and identifier to its agent (Ai) .

TTP$\rightarrow$ P_0: SIG_{TTP} (P_0, h(C,S), Ai)

The migration phase then begins by sending two related messages by each platform except the initial platform P_0 , which does not send a message back to the TTP. The first message is sent to the next platform and the second is sent to the TTP:

$Pi \rightarrow Pi+1$: Pi, P_0,TTP, SIGpi ($Pi+1$, ENCk(C, S), {O_o, O_1, O_2…,Oi}),tpi), ENCpi+1(k), SIG_{TTP} (P_0, h(C,S), Ai)

Immediately upon receipt of the agent, the receiver platform *Pi* sends a receipt message to the trusted third party *TTP*, indicating that he received a mobile agent from the sender and to verify to the TTP that the code and state were received correctly:

Pi→*TTP*: SIGpi (*Pi*, *Pi*-1, *Pi*+1, Ai, h { O_o, O_1,O_2…,Oi})

For the first message, the first three fields are needed to determine whose public keys need to be retrieved. The first field indicates the sender of the message to verify the fourth field, the second field is used to inform the receiver who is the agent originator to encrypt the order offer to ,and TTP identity is used to authenticate the last field .

The fourth field is signed by the sender of the message to verify the sender. This field contains the identity of the recipient P_{i+1}.The recipient examines this field and checks if the message was intended for it. The code C, state S is encrypted using digital envelope scheme in the fifth field. This encryption prevents unauthorized third parties from attempting to execute the mobile agent. The field contains the up to the receiver offers and a timestamp tpi. The recipient uses the timestamp to verify the freshness of the message. The timestamp should contain the time the message is sent and the duration of how long the message is valid.

The final field is signed by the TTP. With the final field *Pi*+1, can verify that the code and state were not tampered with and that it was actually sent by P_0, and authenticated by the TTP. Also from this field, *Pi*+1 can also know the agent identifier. The unique agent identifier links all the messages in the same session together.

The receipt message begins with the second platform since the first one sends only the registration message. These messages are used by the TTP to keep track of the migration path and execution results. The receipt message contains the identity of the sender, previous and next hop along with mobile agent identifier and hash values of up to receiver offers. With this message, TTP can verify that *Pi* received a mobile agent execution request correctly and he is the intended receipt.

The offers are modeled after that used in the publicly verifiable chained digital signature protocol as follows:

$O_0 = ENC_{P0}(o_0, h(r_0, P_1))$
$O_i = ENC_{P0}(o_i, h(O_{i-1}, P_{i+1}))$

The signature part was removed as every platform now signs the up to received offers in the message send to the TTP, the initial encapsulated offer O_0 is slightly different from that of the others. The originator creates an empty initial offer o_o to include in his encapsulated offer O_0. It is empty because it is not offering anything, as it is the originator. The originator generates a random number to use in the hash function because there do not exist a previous encapsulated offer.

6. System Performance

It was explained that the publicly verifiable chained digital signature protocol has a weakness. A malicious platform can truncate the set of offers, modify or replace the code and state, build a new chain of offers with the new code and state, and append the new offers to the truncated set of offers. To verify the security provided by the designed protocol, the

following examples shows how the truncation can simply be detected .Let us assume that the mobile agent is on platform seven hoping to platform eight, the related messages will be:

$P_7 \rightarrow P_8$: P_7, P_0,TTP, $SIGp_7$ (P_8,ENCk(C,S),{ O_0,O_1,O_2, O_3,O_4,O_5, O_6,O_7}),tp_7), ENC_8(k), SIG_{TTP} (P_0, h(C,S), Ai)

$P_7 \rightarrow TTP$: $SIGp_7$ (P_7, P_6, P_8, Ai, h { O_0,O_1,O_2, O_3,O_4,O_5, O_6,O_7})

Now if P_7 wants to truncate the set of offers say from O_4,O_5, O_6 as follows :
$P_7 \rightarrow P_8$: P_7, P_0, $SIGp_7$ (P_8,ENCk(C,S),{ O_0,O_1,O_2, O_3, O_7}),tp_7), ENC_8(k), SIGpo(h(C,S), Ai, TTP)

$P_7 \rightarrow TTP$: $SIGp_7$ (P_7, P_6, P_8,Ai, h{ O_0,O_1,O_2, O_3, O_7})

First the truncation can be discovered by the mobile agent originator from the set of offers Since the offer O_3
$O_3 = ENC_{P0}(o_3, h(O_2, P_4))$

But rather than discovering that when the mobile agent returns to the home platform, the truncation could be discovered by the TTP using the registered mobile agent path as follows:
$P_2 \rightarrow TTP$: $SIGp_2$ (P_2, P_1, P_3, Ai, h {O_0, O_1, O_2})
$P_3 \rightarrow TTP$: $SIGp_3$ (P_3, P_2, P_4, Ai, h {O_0, O_1, O_2, O_3})
$P_4 \rightarrow TTP$: $SIGp_4$ (P_4, P_3, P_5, Ai, h {O_0, O_1, O_2, O_3 ,O_4 })
$P_5 \rightarrow TTP$: $SIGp_5$ (P_5, P_4, P_6, Ai, h {O_0, O_1, O_2, O_3 , O_4 ,O_5 })
$P_6 \rightarrow TTP$: $SIGp_6$ (P_6, P_5, P_7, Ai, h {O_0, O_1, O_2, O_3, O_4 , O_5 ,O_6 })

The TTP can inform the home platform that there was an error to stop the mobile execution or simply ignore the results when it come back. Now let us assume that two platforms now are malicious for example, the mobile agent had already visited platforms P_1, P_2, P_3, P_4,P_5, P_6, P_7, and P_8 in that order and just arrived platform P_9. If platform P_9, who is malicious, suspects that any of the previous offers are better than his, he can collude with his conspirator P_3 and remove offers P_4 through P_8, inclusive. To do this P_9 sends the mobile agent to P_3. P_3 in turn truncates the set of encapsulated offers from his offer O_3, leaving the set {O_0, O_1, O_2}. Then P_3 creates a new encapsulated offer, this time with P_9's as next hope, and sends the mobile agent to P_9. P_9 can then submit his offer and forward the mobile agent to another platform or back to the originator. If the protocol do not involves the usage of the TTP such attack can simply be successful depending only on the chaining relationship, but its now can be detected by the TTP using the mobile agents registered path .

As aforementioned the protocol do not completely rely on the trusted third parity for the integrity, the TTP can solve any future disagree that may occur and, thus, it provides both agent and host security. It provides host security as it receives a copy of the mobile agent signed by the originator; the originator cannot simply provide fake agent or denying sending this agent. The TTP also keeps a copy of the agent execution history each platform send, in case that the originator wants to assure the integrity of the offers he can compare the received offers with the one kept in the TTP as explained above.

Modification of the code and state can be detected. If a malicious platform intends to modify or replace the code and state, he will need to update the hash value in the last field. Having modified that field field, the malicious platform needs to sign it with his own private key, as he cannot sign it with the TTP's private key, the receiving platform can deduce that the sender was malicious. The digital envelope provides basic advantages. The scheme provides the integrity of the whole of the message two components, provides most of the computational speed advantage of the secret key methods, with the key management security advantages of the much slower public key schemes.

7. Conclusion

This paper presents a new protocol, for protecting mobile agents and their data. The used encryption and signature mechanisms protect the integrity of the agent itself and ensures the confidentiality of the agent's data and (binary) code.

To ensure that the itinerary is traversed in the correct order, non-repudiation protocol is executed whenever a mobile agent is migrated from one platform to the next; thus; the owner can verify that the agent has followed the itinerary as expected. Obviously; this solution assumes the existence of a Trusted Third Party

The inclusion of the TTP prevents the truncation of offers by protecting the integrity of the migration path and makes the system have no single point of failure. The used digital envelopes provides the integrity of platforms communication message, help solve the key management problem, and increase performance relative to using a public-key system for direct encryption of message data without sacrificing security.

8. References

1. Hamed Aouadi and PR Mohamed Amed ,***"Security Enhancements for Mobile Agents Platforms,*** IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.7B, July 2006.

2. Junqi Zhang, Yan Wang and Vijay Varadharajan, ***"Mobile Agent and Web Service Integration Security Architecture"***, Journal of Communication and Computer, ISSN1548-7709, Volume 4, No.3 ,USA ,2007.

3. Rahual Jha ***,"Mobile agents for E-commerce",***Thesis,KR School of Information Technology Indian Institute of Technology, Bombay , 2002.

4. Dejan Milojicic. ***"Mobile Agent Applications"***. IEEE Concurrency, Volume.7, No.3 pp.80-90, 1999.

5. Volker Roth. ***"Secure Recording of Itineraries through Co-operating Agents"***. In Proceedings of the ECOOP Workshop on Distributed Object

Security and 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations, p.147-154, INRIA, France, 1998.

6. W. Jansen and T. Karygiannis. ***"Mobile Agent Security"***.In NIST Special Publication 800-19. National Institute of Standards and Technology, 2000

7. 7. Stefan Funfrocken. ***"Protecting Mobile Web-Commerce Agents with Smartcards"***. In Proceedings of the First International Symposium on Agent Systems and Applications / Third International Symposium on Mobile Agents (ASA/MA'99), IEEE Computer Society, p. 90-102. 1999.

8. Fritz Hohl. ***"Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts"***. In Giovanni Vigna, editior, Mobile Agents and Security, LNCS 1419, p.92-113. Springer, 1998.

9. Sau-Koon Ng and Kwok-Wai Cheung. ***"Protecting Mobile Agents against Malicious Hosts by Intention Spreading"***. In H. Arabnia, editor, Proc.Int.Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Vol 2, CSREA, pp.725-729, 1999.

10. G.Karjoth,N.Asokan and C.Gulcu***."Protecting the computation results of free –roaming agents."***Proceeding of the Second International Workshop,Mobile Agents,Springer,1998.

11. Volker Roth ***"On the Robustness of some Cryptographic Protocols for Mobile Agent Protection."*** Proceedings of Mobile Agents 2001. Springer-Verlag Lecture Notes in Computer Science, vol. 2240. December 2001.